المراجعة النهائية ـ للصف الثالث الإعدادي

الفصل الأول : البيانات Data

عند إدخال البيانات يتم تخزينها موقتًا في ذاكرة الكمبيوتر المؤقتة (RAM).

للتعامل مع البيانات (معالجة البيانات) في ذاكرة الكمبيوتر (RAM) لابد أن يحدد لها **اسم معين ونوع.**

وكل بيان يحتاج إلى مساحة تخزين معينة حسب نوعه.

أنواع البيانات التي تتعامل معها لغة VB.Net :

- البيانات **الرقمية** **Numerical** : يتم الإعلان عنها بـ :

بيانات رقمية صحيحة Integer (Long – Integer – Short – Byte)

بيانات رقمية غير صحيحة Non – Integer (Decimal – Double – Single)

- البيانات **الحرفية** **Character** : يتم الإعلان عنها بـ : String – Char

- البيانات **المتنوعة** **Miscellaneous** : يتم الإعلان عنها بـ : Object – Boolean – Date

✓ البيانات المتنوعة لا تندرج تحت تصنيف البيانات الرقمية أو الحرفية.

يتم الإعلان عن بيانات التاريخ والوقت بـ : Date ويوضع التاريخ والوقت بين علامتي **# #**

يتم الإعلان عن البيانات المنطقية بـ : Boolean وتأخذ القيمة True أو False.

البيان من النوع Byte الحد الأدنى له القيمة 0 والحد الأقصى 255.

الثوابت عبارة عن أماكن محجوزة في ذاكرة الكمبيوتر – عند الإعلان عنها لابد أن نحدد للثابت اسم ونوع وقيمة.

✓ قيمة الثابت (ثابتة) لا تتغير أثناء سير البرنامج.

(من الثوابت الرياضية : قيمة (ط) – الثوابت الفيزيائية : عجلة الجاذبية وسرعة الضوء والصوت)

قواعد تسمية المتغيرات والثوابت :

١- يفضل أن يكون الاسم معبرًا عن الغرض منه

٢- أن **يبدأ** اسم الثابت أو اسم المتغير بحرف أو علامة الشرطة السفلي (_)

٣- ألا يحتوي الاسم على أي من **الرموز** أو **العلامات الخاصة** مثل (المسافة ، النقطة . ، ؟ ، ! ، الشرطة - ، * ، + ... إلخ)

٤- ألا تستخدم **الكلمات المحجوزة** للغة VB.Net في تسمية الثوابت والمتغيرات مثل (Single – Double – Dim).

إعلان **الثوابت** في VB.Net : يتم إعلان الثوابت باستخدام الكلمة المحجوزة **Const** وفقًا للصيغة التالية :

Const Constant Name **As** Datatype = **Value**

Const اسم الثابت **As** نوع الثابت = القيمة الثابتة المخزنة



إذا كانت القيمة تاريخًا أو وقتًا فيجب وضعها بين علامتي ## إما إذا كانت **قيمة نصية** فيجب وضعها بين **علامتي تنصيص " "**

إعلان الثوابت هي عملية إعطاء أسماء للخلايا التي تحمل قيم ثابتة.

لتشغيل البرنامج أنقر على زر **Start Debugging** أو **F5** من لوحة المفاتيح.

نطاق (مكان) إعلان المتغير أو الثابت هو الذي يحدد المكان المسموح باستعمال هذا المتغير أو الثابت فيه داخل البرنامج.

يوجد **مستويات** للإعلان عن المتغيرات أو الثوابت (على **مستوي الإجراء Sub** أو **على مستوي التصنيف Class**).

المتغيرات : عبارة عن أماكن محجوزة في ذاكرة الكمبيوتر ويتم تحديد **اسم ونوع** المتغير عند الإعلان عنه.

قيمة المتغير عادة **تتغير** أثناء سير البرنامج (ولذلك سمي بالمتغير)

يمكن أن يأخذ المتغير **قيمة ابتدائية** (تسمي Initial Value) ثم **تتغير** هذه القيمة أثناء سير البرنامج حسب طبيعة البرنامج.

إعلان **المتغيرات** في VB.Net : يتم إعلان المتغيرات باستخدام الكلمة المحجوزة **Dim** وفقًا للصيغة التالية :

Dim Variable Name **As** Datatype [= Initial Value]

Dim اسم المتغير **As** نوع المتغير = (اختياري) القيمة الأولية المخزنة

يقصد بالتخصيص **تعيين قيمة** لمتغير أو لثابت أو لخاصية أداة تحكم.

جملة التخصيص تتكون من طرفين بينهما علامة (**= معامل التخصيص**).

```
Dim x As Integer
X = 10
```

تستخدم جملة التخصيص **لتخزين قيمة** داخل متغير أو ثابت وتتكون من :

الطرف الأيسر(مكان تخزين القيمة) : المتغير أو الثابت الذي سيتم تخزين القيمة فيه

الطرف الأيمن (القيمة) : القيمة التي سيتم تخزينها (قيمة مجردة أو قيمة من متغير أو قيمة من تعبير رياضي أو قيمة من خاصية من مربع نص)

قواعد إضافية عند كتابة الكود :

تستخدم **&** (معامل الربط) للفصل بين كل متغير وآخر أو لربط النصوص.

تستخدم الشرطة السفلية (_) لكتابة الكود على أكثر من سطر في حالة إذا كان سطر الكود طويل وذلك لتنظيم وتسهيل قراءته

تستخدم **vbCrlf** لإنشاء سطر جديد في الكود.

تعبر **Me** عن نافذة النموذج Form الحالية.

يمكن إضافة تعليقات أو ملاحظات باستخدام **REM** أو علامة الاقتباس المفردة (**'**) لتوضيح الكود.



أولويات العمليات الحسابية : عند تنفيذ العمليات الحسابية تتبع لغة VB.Net قواعد معينة :

```
Dim result As Integer
result = ( 5 + 3 ) * 2
```

١- الأقواس ( ) يتم تنفيذ العمليات داخل الأقواس أولًا (من الداخل إلى الخارج).

٢- الأسس (^)

٣- الضرب (*) والقسمة (/) : يتم تنفيذها من اليسار لليمين وفقًا للأسبقية.

٤- الجمع (+) والطرح (-) : يتم تنفيذها من اليسار لليمين وفقًا للأسبقية.

أنواع الأخطاء في VB.Net : يمكن تصنيف الأخطاء البرمجية إلى ٣ أنواع رئيسية :

١- أخطاء **لغوية** (Syntax Errors) : هي أخطاء في الصيغة العامة لأوامر اللغة وتحدث عند كتابة الكود بشكل غير سليم.

٢- الأخطاء **المنطقية** (Logic Errors) : تحدث بسبب صياغة تعبيرات حسابية أو منطقية بصورة خطأ تؤدي إلى نتائج خطأ

٣- أخطاء **أثناء التشغيل** (Runtime Errors) : تحدث عند تخصيص قيمة أصغر أو أكبر من المدي المسموح به لنوع البيان المستخدم.

الفصل الثاني : التفرع

التفرع باستخدام جملة **If ... Then** : تستخدم عندما يكون هناك **شرط واحد فقط** وبديل واحد فقط يجب التحقق منه إذا تحقق الشرط (True) يتم تنفيذ الكود (الأكواد) الذي يلي Then لحد End If وإذا لم يتحقق يتم تجاهل الأكواد وينتقل إلى الكود الذي يلي End If. (تنتهي جميع جمل If بـ End If)

```
If Condition Then
Code1
End If
Code2
```

الصيغة العامة :

التعبير الشرطي (Conditional Expression) : جزء من كود البرمجة

وله ناتج منطقي (True أو False) ويتكون من ثلاثة أجزاء رئيسية :

١- معامل المقارنة : < - > - =< - => - <> - =

٢- المعامل الأول : قيمة – متغير – ثابت – تعبير حسابي.

٣- المعامل الثاني : قيمة – متغير – ثابت – تعبير حسابي يقارن مع المعامل الأول.

التفرع باستخدام جملة **If ... Then ... Else** : تستخدم عندما يكون هناك شرط واحد فقط **واحتمالان** مختلفان للتنفيذ.

```
If Condition Then
Code1
Else
Code2
End If
Code3
```

إذا كان الشرط (True) ⟵ يتم تنفيذ الكود داخل Then

ويتم تجاهل الكود داخل Else.

وإذا كان الشرط (False) ⟵ يتم تنفيذ الكود داخل Else

ويتم تجاهل الكود داخل Then.

التفرع باستخدام جملة **Select ... Case** : تستخدم عندما **يعتمد التفرع على قيمة متغير** واحد وهناك **عدة شروط** محتملة

الصيغة العامة :

```
Select Case Variable
Case1 Value1
Code1
Case2 Value2
Code2
Case Else
Code3
End Select
```

تبسط جملة **Select ... Case** الكود مقارنة باستخدام If ... Then ... Else مما يجعله أوضح وأسهل قراءة.

توفر جملة **Select ... Case** العديد من الأكواد ويفضل استخدامها عندما يكون التفرع لأكثر من شرطين.

يمكن الاستغناء عن **Case Else** طالما لا حاجة لها.

يتم تنفيذ **Case Else** فقط إذا لم تتطابق أي من القيم المحددة.

إثرائي

أداة Listbox تتيح لك أن تنشأ وتعرض قائمة من العناصر (الصفوف) يمكن لمستخدم البرنامج أن يختار منها عنصر أو أكثر ويتم إضافة عناصر القائمة Listbox في نمط التصميم عن طريق الخاصية Items.

Index تشير إلي رقم العنصر داخل الأداة – ترتب Index العناصر داخل الأداة Listbox حيث يبدأ من الرقم صفر والعنصر الثاني يأخذ الرقم ١ والعنصر الثالث يأخذ الرقم ٢ وهكذا ... إلخ – الحدث الافتراضي للأداة Listbox هو SelectedIndexChanged (عند تحديد المستخدم عنصر آخر بدلًا من المحدد الحالي يتغير بالتالي رقم العنصر حسب ترتيبه) – الخاصية SelectedIndex تستخدم لمعرفة رقم العنصر (الصف) الذي اختاره المستخدم.

دالة Mod : تستخدم لإيجاد باقي القسمة

تستخدم (" ")Msgbox لعرض رسالة للمستخدم تكتب الرسالة بين القوسين بين علامتي التنصيص.

تستخدم دالة ( )Focus لتركيز مؤشر الفارة في أداة صندوق النص " Textbox "

يمكن تفريغ محتوي Textbox باستخدام " " = Textbox.Text

لاحظ : المعاملات الحسابية هي الطرح (-) والجمع (+) والضرب (*) والقسمة (/) تختلف عن معاملات المقارنة.



الفصل الثالث : التكرار والإجراءات

جملة **For ... Next** هي أحد جمل التكرار المحدود

تستخدم لتكرار كود معين عدد محدد من المرات (**عدد مرات تكرار معروف مسبقًا**).

ينتهي التكرار عندما تكون قيمة متغير العداد أكبر من قيمة نهاية الحلقة التكرارية.

استخدام الأمرStep في جملة التكرار (اختياري) وكذلك وكتابة اسم متغير العداد بجوار الأمر Next.

في حالة عدم استخدام Step فإن القيمة الافتراضية لزيادة المتغير العداد موجب ١ .

إذا كانت قيمة الزيادة موجب ١ فإنه يمكن الاستغناء عن كتابة Step Add Value

بعد Step يكتب مقدار الزيادة (عدد صحيح أو عدد عشري / عدد سالب أو عدد موجب / متغير عدد صحيح او عشري)

كلمة Step تستخدم للتحكم في مقدار زيادة المتغير العداد في كل مرة من مرات التكرار

يمكن جعل قيمة البداية أكبر من قيمة النهاية وتكون قيمة زيادة متغير العداد بالسالب.

```
For Variable = Start Value To End Value Step Increment
Code الأوامر المطلوب تنفيذها
Next Variable
```

القيم الديناميكية (عندما تكون القيم الإبتدائية والنهائية ومعدل الزيادة متغيرات بدلًا من أرقام ثابتة).

جملة التكرار **Do While ... Loop** :

تستخدم لتكرار كود معين عدد من المرات (**غير معروف عدد مرات التكرار مسبقًا**) بناءًا على شرط معين.

(غير معروف نهايته طالما أن شرط تنفيذ التكرار True ويتوقف التكرار عندما يصبح شرط تنفيذ التكرار False).

يتم تنفيذ الكود الذي بين بداية الحلقة التكرارية (Do While) ونهايتها (Loop) طالما أن التعبير الشرطي صحيح True.

يتوقف تنفيذ الكود عندما تصبح قيمة التعبير الشرطي خطأ False ويتم الخروج من الحلقة التكرارية وتنفيذ الكود الذي يلي Loop إن وجد.

```
Do While ( Conditional Expression )
Code الأوامر المطلوب تنفيذها
Loop
```

- ✓ عند إضافة نافذة نموذج Form جديدة ينشأ تصنيف **Class** جديد باسم Form1.

في نطاق هذا التصنيف نعلن عن :

إجراءات الأحداث Event Procedures – المتغيرات Variables – الثوابت Constants.



الإجراء Procedure : هو **مجموعة من الأوامر والتعليمات** Code تحت اسم ، يمكن استدعاءه بهذا الاسم ليتم تنفيذ هذه الأوامر والتعليمات.

فوائد استخدام الإجراءات :

- تجنب تكرار الكود : بدلًا من كتابة نفس الكود عدة مرات نعرف الإجراء مرة واحدة ونستدعيه عند الحاجة.
- تحسين وضوح وصيانة الكود : الإجراءات تجعل الكود أكثر تنظيمًا وأسهل في التعديل.
- إعادة استخدام الكود : يمكن استدعاء الإجراء من أي مكان داخل البرنامج

الإعلان عن الإجراءات : يتم الإعلان عن الإجراء Sub إذا كان لدينا مجموعة من الأوامر يتكرر استخدامها في أكثر من موضع داخل التصنيف. – يتم الإعلان عن الإجراء مرة واحدة ، ويستدعي أي عدد من المرات.

يتم الإعلان عن الإجراء لعدم تكرار كتابة نفس الكود في أكثر من موضع.

يمكن استدعاء الإجراء بكتابة اسمه في أي مكان عندما تريد داخل نافذة الكود.

أنواع الإجراءات في لغة VB.Net : ١- إجراء فرعي Sub (لا يعود بقيمة) ٢- دالة Function (يعود بقيمة)

أسباب الإعلان عن الإجراء **Sub** : وجود كود معين سيتكرر كتابته في أكثر من مكان داخل التصنيف Class

صيغة الإعلان عن الإجراء :

Sub + اسم الإجراء + (وسائط الإجراء parameters)
Code الأوامر المطلوب تنفيذها
End Sub نهاية الإجراء

المعاملات Parameters اختياري :

القيم التي سيتم استخدامها داخل الإجراء

يمكن أن يحتوي الإجراء على أكثر من معامل واحد – عند استدعاءه يجب تمرير قيم Arguments تطابق المعاملات.

الدالة Function : مجموعة من الأوامر تحت اسم معين ، يفضل أن يكون الاسم معبرًا عن وظيفتها ، يتم تطبيقها على مدخلات أو وسائط Parameters وتعود بقيمة.

الدوال Functions : تستدعي الدالة فتعود بقيمة في ضوء القيم المخصصة لها.

Function + اسم الدالة + (وسائط الإجراء parameters) **As** Datatype
Code الأوامر المطلوب تنفيذها
Return Value
End Function

المعاملات Parameters اختياري : القيم المستخدمة داخل الدالة

Datatype نوع البيانات التي سترجعها الدالة – Return القيمة الراجعة : تحدد القيمة التي سترجعها الدالة

يجب أن ترجع الدالة قيمة باستخدام Return

يتم استدعاء الدوال مثل المتغيرات وترجع القيم (تتغير) بناءا على المدخلات

المتغيرات Variables : يمكن تخصيص قيمة لها أثناء الإعلان عنها في نمط التصميم أو أثناء سير البرنامج وكذلك يمكن استخدام القيم المخزنة بها.

الثوابت Constants : لابد من تخصص قيمة لها أثناء الإعلان عنها فقط وكذلك يمكن استخدام القيم المخزنة بها.

الفصل الرابع : التعدي الإلكتروني

الإنترنت لها جانبين : جانب **إيجابي** (نتعلم – نتثقف – نتسلى – نتواصل ونتحاور) وجانب **سلبي** حيث توجد مخاطر عديدة :

- يمكن أن نحصل على معلومات خطأ – انتهاك الخصوصية – انتحال الشخصية
- يمكن أن نقع فريسة لبعض المعتدين عبر وسائط الاتصال الإلكترونية
- سرقة حسابك على مواقع التواصل الاجتماعي مثل موقع التواصل Facebook أو البريد الإلكتروني Email
- تعرض جهازك للفيروسات أو لبرامج التجسس أو لبرامج القراصنة.

التعدي الإلكتروني : هو **سلوك عدواني متعمد** من شخص لآخر عبر وسائط الاتصال الإلكترونية.

صور التعدي الإلكتروني : التحرش – المضايقة – الإحراج – التخويف – التهديد – الابتزاز

الوسائط الإلكترونية للتعدي : عبارة عن **التقنيات** التي يستخدمها المتعدي الإلكتروني وهي كثيرة منها :

١- البريد الإلكتروني Email ٢- المنتديات الإلكترونية Forums ٣- الرسائل الفورية Instant Message

٤- المدونات الإلكترونية Blogger ٥- مواقع التواصل الاجتماعي مثل Facebook

من **أشكال** التعدي الإلكتروني :

١- **التخفي** الإلكتروني : هو استخدام **أسماء مستعارة** تخفي الشخص المعتدي الإلكتروني بغرض عدم كشف أمره والإفلات من العقاب.

٢- **المضايقات** الإلكترونية : هي **رسائل عدائية موجهة** ضد شخص أو أكثر.

٣- **الملاحقة** الإلكترونية : هي شكل من **أشكال المضايقات** الإلكترونية لكن **بشكل متكرر** حيث **يتتبع** المعتدي شخص معين في كافة الوسائط الإلكترونية ويلاحقه

٤- **السب** أو **القذف** الإلكتروني : هو **نشر كلمات عدائية ومبتذلة** ضد شخص أو أكثر من خلال أحد وسائط الاتصال الإلكترونية.

٥- **التشهير** الإلكتروني : هو **نشر معلومات** عن شخص محدد أو أكثر بشكل مسيء

٦- **الاستثناء** الإلكتروني : عبارة عن **تجاهل شخص** أو أكثر من خلال وسائط إلكترونية.

٧- **التهديد** الإلكتروني : هو **إرسال رسائل** إلكترونية تحمل **تهديد أو وعيد** لشخص أو أكثر.

كيف تحمي نفسك من التعدي الإلكتروني : بإتباع قواعد الاستخدام الآمن للإنترنت التالية :

١- لا تشارك أحد بكلمة السر ٢- عدم نشر أي بيانات خاصة ٣- إعداد كلمة مرور يصعب استنتاجها

٤- عدم حذف رسائل المتعدي ٥- عدم مقابلة أحد تعرفت عليه من خلال الإنترنت.

٦- إطلاع ولي الأمر بمن يضايقك عند استخدام الإنترنت.

٧- إنزال البرامج من الإنترنت يكون بإذن معلمك أو ولي أمرك.

٨- عدم إرسال رسائل إلكترونية وأنت في حالة غضب.

